


Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
«Институт космических и информационных технологий»
Кафедра «Информационные системы»

УТВЕРЖДАЮ

Зав. кафедрой ИС

 С. А. Виденин
«__» ____ 2016 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Решение задачи классификации в условиях неполной информации

Руководитель

 16.06
подпись, дата

А. А. Корнеева
инициалы, фамилия

Выпускник

 15.06
подпись, дата

А. Н. Хохлов
инициалы, фамилия

Нормоконтролер

 17.06
подпись, дата

Л. С. Троценко
инициалы, фамилия

Красноярск 2016

Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
«Институт космических и информационных технологий»
Кафедра «Информационные системы»

УТВЕРЖДАЮ
Зав. кафедрой ИС
_____ С. А. Виденин
«___» _____ 2016 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Решение задачи классификации в условиях неполной информации

Руководитель

подпись, дата

А. А. Корнеева
инициалы, фамилия

Выпускник

подпись, дата

А. Н. Хохлов
инициалы, фамилия

Нормоконтролер

подпись, дата

Л. С. Троценко
инициалы, фамилия

Красноярск 2016

РЕФЕРАТ

Выпускная квалификационная работа по теме «Решение задачи классификации в условиях неполной информации» содержит 52 страницы текстового документа, 57 иллюстраций, 4 таблицы, 9 формул, 1 приложение, 11 использованных источников.

КЛАССИФИКАЦИЯ, КЛАСТЕРИЗАЦИЯ, МАШИННОЕ ОБУЧЕНИЕ, АНАЛИЗ ДАННЫХ, РАСПОЗНАВАНИЕ ОБРАЗОВ, ТЕХНИЧЕСКАЯ ДИАГНОСТИКА

Цель работы: повышение точности решения задачи распознавания образов при компьютерной диагностике электрорадиоизделий.

Задачи работы:

- реализовать и исследовать алгоритмы классификации: k-ближайших соседей, FRiS-STOLP;
- реализовать и исследовать алгоритмы кластеризации: FOREL, k-средних, кластеризатор на основе самоорганизующейся сети Кохонена;
- проанализировать особенности работы алгоритмов на модельных и реальных данных, сравнить полученные результаты;
- разработать автоматизированную систему для решения задачи распознавания образов.

Основные результаты, связанные с исследованием и разработкой алгоритмов, получены автором лично. Постановка задач и обсуждение возможных путей их решения осуществлялись совместно с руководителем.

Актуальность и практический аспект проблемы технической диагностики связаны с широким распространением практических задач в жизни общества, решение которых без дополнительных разработок и исследований невозможно. Компьютерная диагностика является основой надежности, безопасности и эффективности промышленных и строительных объектов.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Задача распознавания образов в технической диагностике	6
1.1 Задача интеллектуального анализа данных – Data Mining	6
1.2 Задачи классификации и кластеризации.....	8
1.3 Задача технической диагностики	10
1.4 Обзор решаемых задач	11
1.5 Выбор программных средств разработки.....	17
1.6 Выводы	21
2 Обзор методов классификации и кластеризации.....	22
2.1 Метод k ближайших соседей	22
2.2 Алгоритм FRiS-STOLP	23
2.3 Алгоритм FOREL	26
2.4 Метод k-средних.....	32
2.5 Кластеризатор на основе нейронной сети Кохонена.....	34
2.6 Выводы	40
3 Численные исследования	41
3.1 Задача 1 – «Ирисы Фишера».....	41
3.2 Задача 2 – одна партия ЭРИ.....	44
3.3 Задача 3 – несколько партий ЭРИ	46
3.4 Выводы	50
ЗАКЛЮЧЕНИЕ	51
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	52
ПРИЛОЖЕНИЕ А Техническое задание	53

ВВЕДЕНИЕ

В современном мире на первый план выходят компьютерные технологии, информационные системы, управление и прочие направления, важные для развития новых технологий. Быстрый ритм жизни способствует разработке многофункциональной продукции, которая значительно упрощает нашу жизнь. Именно поэтому наша повседневная жизнь сильно зависит от бытовых приборов, техники и прочих приспособлений, автоматизирующих трудоемкие процессы.

Под технической диагностикой понимают определение технического состояния устройства. Актуальность выбранной темы обусловлена значимостью технической диагностики перед эксплуатацией каких-либо механизмов или микросхем. В работе рассматривался вопрос технической диагностики электрорадиоизделий (ЭРИ).

На сегодняшний день существует множество предприятий, осуществляющих анализ отказов и выявление скрытых дефектов ЭРИ и полупроводниковых приборов. Проведение контроля качества ЭРИ позволяет обнаружить изделия с отклонениями или обеспечить гарантию их работы.

На территории России имеется несколько испытательных центров ракетно-космической промышленности, занимающихся испытаниями ЭКБ для бортовой аппаратуры. Они взаимодействуют с производителями и поставщиками, а также участвуют при разработке методик входного контроля. Одним таких центров Красноярского края является Железногорское ОАО «ИТЦ – НПО ПМ».

В компьютерной диагностике большую роль играет обработка данных. В связи с большим количеством исследуемых данных их обработка становится все более затруднительной. Поэтому разрабатываются специальные методы, позволяющие работать с многомерными наблюдениями. К одним из таких способов относят задачи классификации и кластеризации.

Задача классификации относится к разделу обучения с учителем. Для задач данного типа характерно контролируемое обучение с присутствием обучающего множества с указанной принадлежностью объектов к классам. Задача кластеризации относится к разделу обучения без учителя. Для задач данного типа характерно неконтролируемое обучение, когда принадлежность объектов к каким-либо классам не указана. Множество объектов необходимо разделить на один или более кластеров данных.

Цель работы: повышение точности решения задачи распознавания образов при компьютерной диагностике электрорадиоизделий.

Задачи работы:

- реализовать и исследовать алгоритмы классификации: k-ближайших соседей, FRiS-STOLP;
- реализовать и исследовать алгоритмы кластеризации: FOREL, k-средних, кластеризатор на основе самоорганизующейся сети Кохонена;
- проанализировать особенности работы алгоритмов на модельных и реальных данных, сравнить полученные результаты;
- разработать автоматизированную систему для решения задачи распознавания образов.

Выпускная квалификационная работа состоит из введения, трёх глав, заключения, списка используемых источников, приложения.

В первой главе раскрываются теоретические основы предметной области: определяются понятия и основные аспекты задач классификации и кластеризации, освещаются способы их решения. Также рассматриваются задачи, решаемые в ходе выполнения работы, приведены иллюстрации с распределениями массивов данных по различным срезам, обосновывается выбор программных средств разработки.

Во второй главе исследуются особенности разрабатываемых алгоритмов, дается краткое описание алгоритма, используемые параметры, коэффициенты и входные данные, описывается порядок выполнения алгоритма, приводятся примеры работы алгоритма на модельных данных с иллюстрациями, определяются основные достоинства и недостатки рассматриваемых методов.

В третьей главе приводится подробное решение исходных задач с иллюстрациями, таблицами результатов и сравнением полученных данных различными алгоритмами при использовании разработанной автоматизированной системы.

Данная работа является исследовательской и может быть полезна при решении задачи технической диагностики.

1 Задача распознавания образов в технической диагностике

1.1 Задача интеллектуального анализа данных – Data Mining

На протяжении всей жизни человек познает большое количество информации. В связи с развитием компьютерных технологий, появлением интернета, различных баз данных на людей обрушились огромные информационные потоки. Для восприятия и понимания этих данных возникло такое направление как Data Mining.

Data Mining (рус. добыча данных, интеллектуальный анализ данных) – технология, используемая для обнаружения в исходных данных заранее неизвестных полезных знаний. Термин был введен Григорием Пятецким-Шапиро в 1989 году.

Data Mining применяется в самых различных сферах человеческой деятельности: для реализации масштабных проектов в бизнесе, промышленности, геологии, медицине и других областях.

Предполагается, что существует некоторая база данных, в которой имеют место «скрытые знания». Под ними понимают какую-то новую информацию, нетривиальные знания, обнаружить которые не так просто. Они обязательно должны иметь практическую ценность для выполнения поставленной цели. Такие знания могут быть представлены в наглядной форме и быть легко интерпретированы.

В основе Data Mining лежит простая схема применения:

- постановка задачи;
- предварительная обработка данных: сбор, подготовка, внедрение;
- построение модели, оценка точности;
- применение разработанной модели.

Data Mining позволяет решить задачи, которые требуют большого количества вычислений и не могут быть решены без помощи компьютера. Высокий уровень развития современных технологий позволяет значительно упростить обработку данных и сократить время работы алгоритмов.

Основные алгоритмы Data Mining

Классификация – один из разделов машинного обучения, который заключается в следующем: имеется множество объектов с известными признаками и принадлежностью к какому-либо из классов, а также набор объектов с похожими параметрами, класс которых необходимо определить на основе имеющейся информации.

Пример задачи классификации изображен на рисунке 1, объекты в форме треугольника и квадрата принадлежат разным классам, принадлежность круглых объектов необходимо определить.

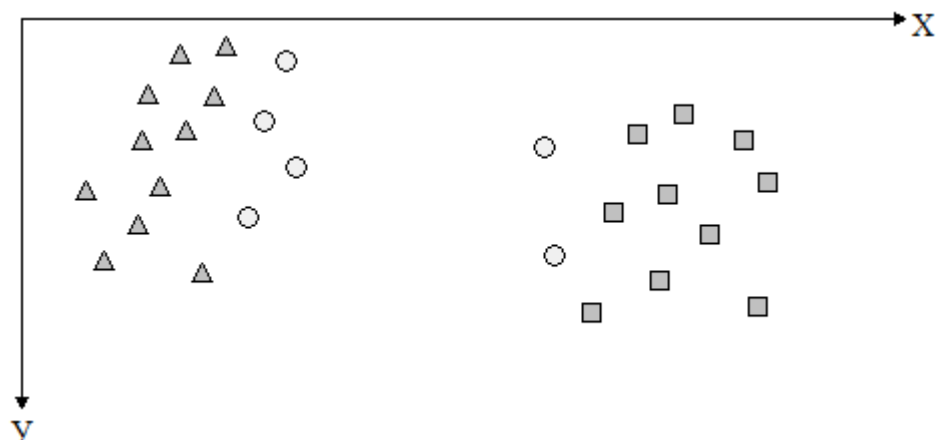


Рисунок 1 – Задача классификации

Кластеризация – один из разделов машинного обучения, где имеется массив объектов с известными признаками, которые необходимо разделить на отдельные непересекающиеся множества, называемые кластерами. На рисунке 2 представлен пример задачи кластеризации и ее решение.

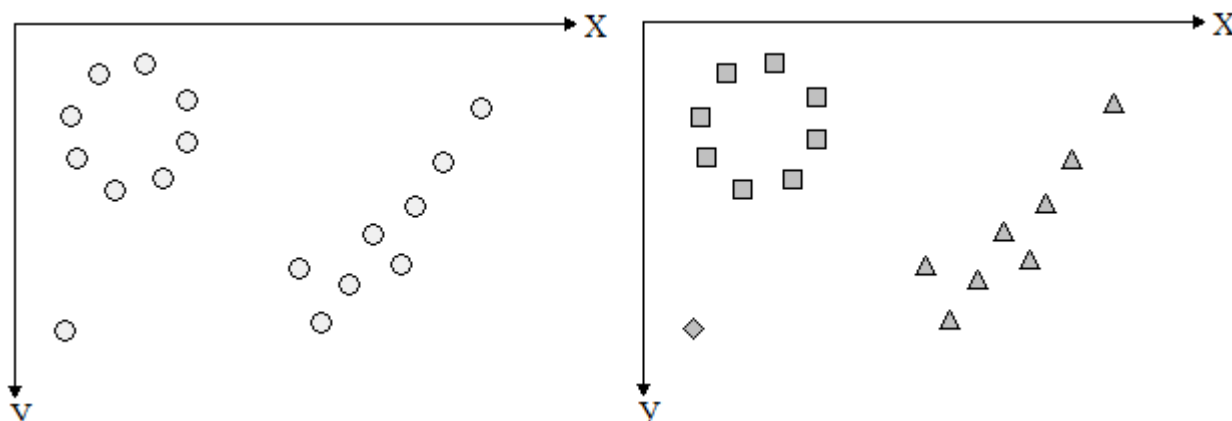


Рисунок 2 – Задача кластеризации

Регрессия – метод для поиска или восстановления зависимости между отдельными переменными для исследования их влияния на зависимую переменную или результат. На рисунке 3 представлен пример задачи регрессии и ее решение.

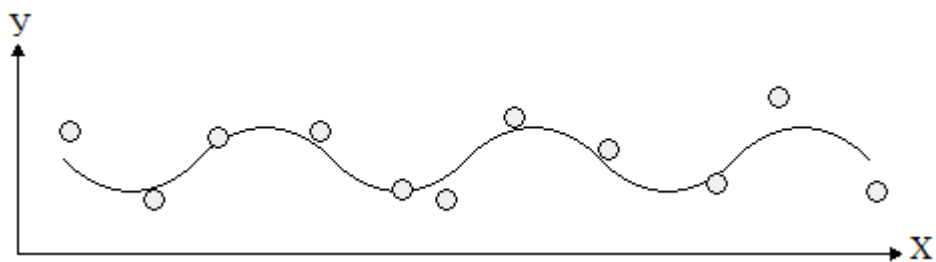


Рисунок 3 – Задача регрессии

Ассоциация – выявление закономерностей между взаимосвязанными событиями. Если из события А следует событие В, то такое правило называется ассоциативным.

Анализ отклонений – выявление данных, заметно отличающихся от общего шаблона, что позволяет сократить объем информации для выявления главных компонент.

Генетические алгоритмы – методы, основанные на эволюции поколений итеративным способом с использованием механизмов, аналогичных естественному отбору в природе. Основными этапами являются:

- создание начальной популяции;
- отбор;
- выбор родителей;
- размножение;
- мутации.

1.2 Задачи классификации и кластеризации

Задача классификации относится к разделу обучения с учителем. Для задач данного типа характерно контролируемое обучение с присутствием обучающего множества с указанной принадлежностью объектов к классам.

Определяемые классы могут быть как пересекающиеся, так и непересекающиеся. Пример таких распределений отображен на рисунке 4.

На сегодняшний день существует большое множество алгоритмов классификации, которые могут давать решения с различной степенью точности. Точной систематизации методов классификации не существует, однако можно выделить основные используемые:

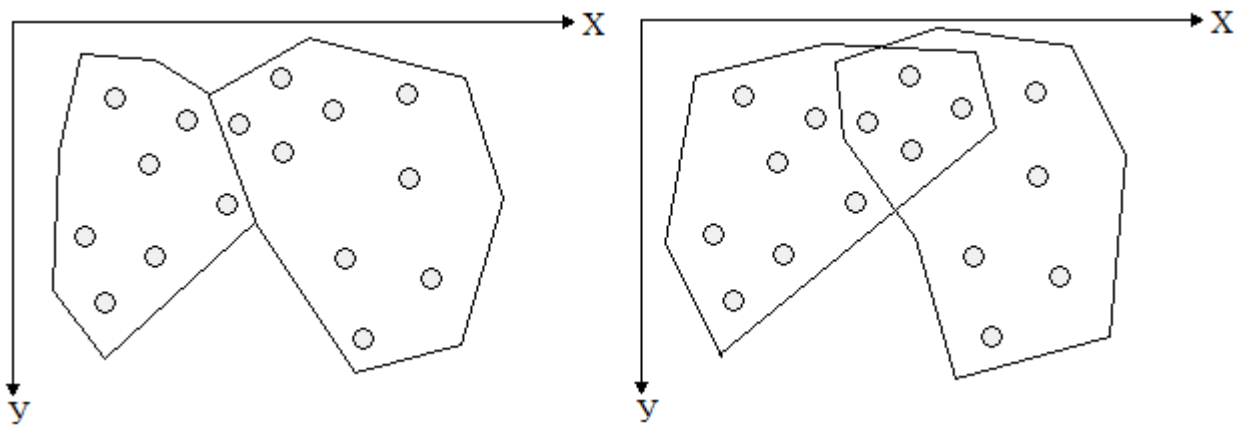


Рисунок 4 – Непересекающиеся и пересекающиеся множества

- деревья решений – средство для поддержки принятия решений. Структура деревьев представляет собой «ветки» и «листья». Первые содержат атрибуты, оказывающие влияние на итоговый результат, а вторые содержат значения целевой функции;
- байесовская классификация – представляет из себя классификатор, основанный на теореме Байеса: если плотности распределения каждого из классов известны, то искомый алгоритм можно записать в одной формулой;
- искусственные нейронные сети – система взаимодействующих между собой нейронов, каждый из которых имеет входную и выходную связь. На вход поступают данные для обучения сети, они влияют на текущее состояние нейрона, что влечет за собой изменение выходных сигналов;
- статистические методы – научные методы анализа статистических данных, полученные в результате наблюдений или экспериментов;
- методы ближайшего соседа – заключаются в нахождении определенного количества наиболее похожих наблюдений каких-либо явлений на основе евклидова расстояния. Найденные объекты используются для определения класса объекта.

Задача кластеризации относится к разделу обучения без учителя. Для задач данного типа характерно неконтролируемое обучение, когда принадлежность объектов к каким-либо классам не указана. Множество объектов необходимо разделить на один или более кластеров данных.

Приведем основные способы решения задачи кластеризации:

- итеративные алгоритмы, основанные на разделении исходной выборки на N классов;
- иерархические алгоритмы. Изначально каждый объект является отдельным кластером. В процессе работы мелкие кластеры объединяются в более крупные;

- методы основанные на концентрации объектов;
- модельные методы, основанные на использовании моделей, наиболее приближенных к искомым данным.

Иногда определенные задачи могут иметь не единственное решение. Это применимо как к задаче классификации, так и кластеризации.

1.3 Задача технической диагностики

Под технической диагностикой понимают определение технического состояния устройства. Техническая диагностика перед эксплуатацией каких-либо механизмов или микросхем имеет большую значимость. В работе рассматривался вопрос технической диагностики ЭРИ. Пример ЭРИ представлен на рисунке 5.

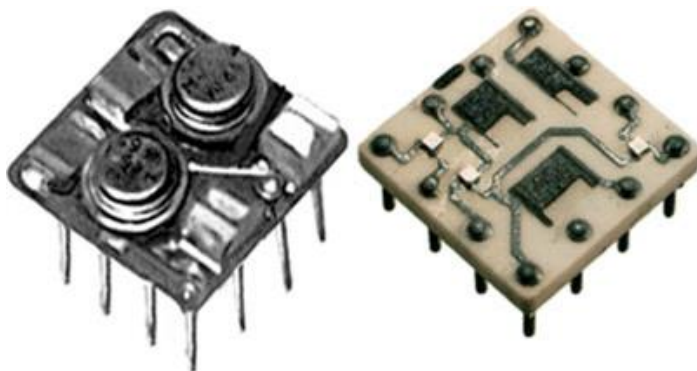


Рисунок 5 – Электрорадиоизделия

На сегодняшний день существует множество предприятий, осуществляющих анализ отказов и выявление скрытых дефектов ЭРИ и полупроводниковых приборов. Проведение контроля качества ЭРИ позволяет обнаружить изделия с отклонениями или обеспечить гарантию их работы.

На территории России имеется несколько испытательных центров ракетно-космической промышленности, занимающихся испытаниями ЭКБ для бортовой аппаратуры. Они взаимодействуют с производителями и поставщиками, а также участвуют при разработке методик входного контроля. Одним таких центров Красноярского края является Железногорское ОАО «ИТЦ – НПО ПМ» (рисунок 6).



Рисунок 6 – Логотип предприятия

Существует три вида испытаний ЭРИ:

- отбраковочные испытания;
- разрушающий физический анализ (РФА);
- диагностический неразрушающий контроль.

Отбраковочные испытания дают подробную информацию о качестве электронной компонентной базы (ЭКБ). Они включают в себя проверку соответствия ЭКБ требованиям документации и позволяют отбраковать потенциально ненадежные изделия.

РФА заключается в проведении физико-технической экспертизы разрушающими методами для выборочных изделий из одной партии. По полученным результатам проводятся дополнительные испытания для партий с отрицательными результатами, и принимаются решения о дальнейшей их эксплуатации.

Диагностический неразрушающий контроль проводится с целью выявления потенциально ненадежной ЭКБ. Его особенностью является то, что отбраковываются изделия, соответствующие нормативно-технической документации, но имеющие признаки скрытых дефектов и низкого качества. Он позволяет выявить дефекты, обнаружить которые невозможно при проведении отбраковочных испытаний или РФА.

1.4 Обзор решаемых задач

Для настройки алгоритмов, определения оптимальных значений параметров в качестве используемой выборки был взят массив данных под названием Ирисы Фишера (от англ. – Fisher's Iris data set), взятые из репозитория UCI задач машинного обучения, содержащего как реальные, так и модельные наборы данных. Это одна из самых известных баз данных, используемых для распознаваний образов. Массив данных состоит из 150 экземпляров цветков ириса, по 50 экземпляров каждого из трех видов:

- возможность написания программ практически любого назначения: утилиты, драйверы, библиотеки и т. д.;
- поддержка различных стилей и технологий программирования: объектно-ориентированное программирование, процедурное программирование, метапрограммирование;
- эффективность: язык предоставляет программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы;
- кроссплатформенность: стандарт языка накладывает минимальные требования на ЭВМ для запуска скомпилированных программ.

На сегодняшний день существует множество различных сред программирования, предоставляющие различный функционал для разработки приложений. В качестве инструментария разработки программного обеспечения была использована программа Qt Creator. Это кроссплатформенная свободная IDE (от англ. Integrated development environment – интегрированная среда разработки) для разработки на C, C++, QML. Она включает в себя визуальные средства разработки интерфейса с использованием QtWidgets: шаблоны контейнеров, средства работы с мультимедиа и т. п., что позволяет упростить разработку приложения.

При использовании библиотеки Qt объектная модель C++ расширяется следующими возможностями:

- механизм сигнал-слот;
- свойства объекта;
- внутренние таймеры;
- манипулирование иерархией наследования классов;
- управляемые указатели;
- динамическое приведение типов.

В состав Qt включены инструменты разработчика с графическим или консольным интерфейсом. В их числе:

- assistant – графическое средство для просмотра гипертекстовой документации по инструментарию и библиотекам Qt;
- designer – графическое средство для создания и сборки пользовательских интерфейсов на основе компонентов Qt;
- qmake – кросс-платформенный генератор Makefile;
- moc – компилятор метаобъектов (обработчик расширений Qt для C++);
- uic – компилятор пользовательских интерфейсов из файлов .ui, созданных в Qt Designer;
- rcc – компилятор ресурсов из файлов .qrc.

Одной из основных причин выбора данной IDE является наличие опыта работы с ней: она неоднократно использовалась как при выполнении курсовых, так и лабораторных работ.

Перед составлением программной реализации системы необходимо определить ее составные части, выбрать необходимые библиотеки для реализации основных функций. В процессе выполнения выпускной квалификационной работы была разработана следующая структурная схема программной системы, состоящая из массива данных, обработчика, основной программы, блока настройки параметров и формирования результатов эксперимента. Структура автоматизированной системы представлена на рисунке 14.

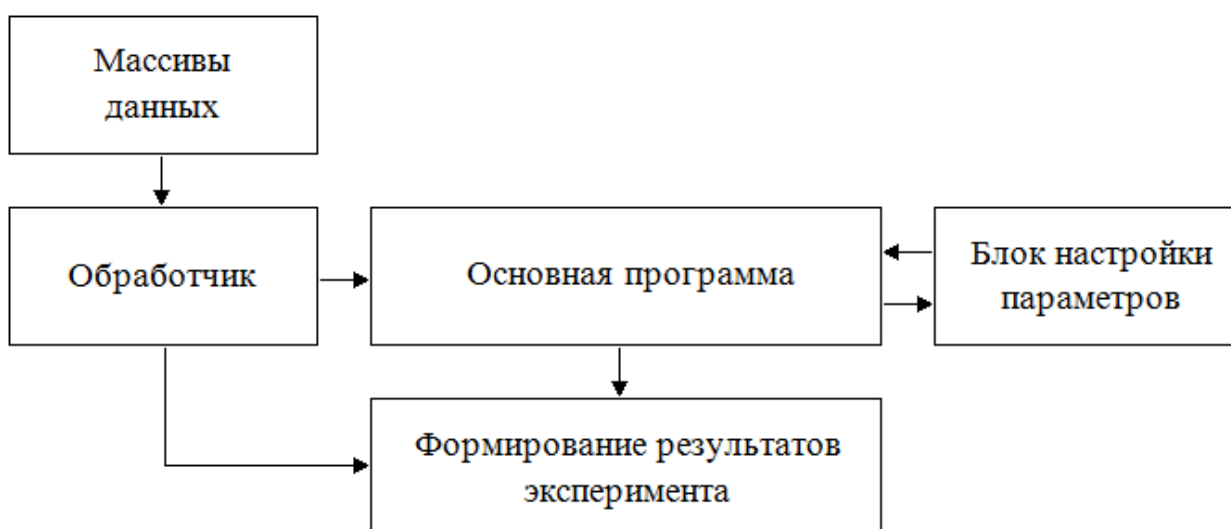


Рисунок 14 – Структурная схема программной системы

Массивы данных

Рациональнее всего использовать файлы формата .txt, так как он наиболее удобен при чтении и записи данных, поэтому данный блок состоит из следующих наборов задач, сохраненных в текстовых файлах:

- модельные задачи классификации;
- реальные задачи классификации;
- модельные задачи кластеризации;
- реальные задачи кластеризации.

Каждая строка текстового файла содержит в себе информацию об отдельных наблюдениях, разделенных запятыми, представленные в виде десятичной дроби. Количество рассматриваемых атрибутов не ограничено и должно быть одинаковым для всех строчек файла.

Обработчик

В связи с большим разбросом значений, данные, пересылаемые в программу, должны быть предварительно обработаны:

- нормализация параметров, чтобы их значения находились в промежутке от 0 до 1;
- приведение бинарных переменных к значению 0 или 1.

Блок настройки параметров.

Поскольку в разрабатываемой программе будет реализовано несколько алгоритмов, каждый из которых обладает своими особенностями, необходимо добавить отдельное окно, которое позволит подобрать оптимальные значения коэффициентов и параметров для правильной работы алгоритмов.

Основная программа и формирование результатов эксперимента

Блок формирования результатов позволяет просмотреть информацию об исходной выборке: принадлежность данных к классам, значение отдельных параметров. Также имеется возможность просмотра гистограммы, отображающей плотность распределения объектов, графика, отражающей распределение данных в каком-либо срезе по двум переменным.

Главное окно программы должно представлять удобный пользовательский интерфейс. На рисунке 15 представлен интерфейс разработанной автоматизированной системы.

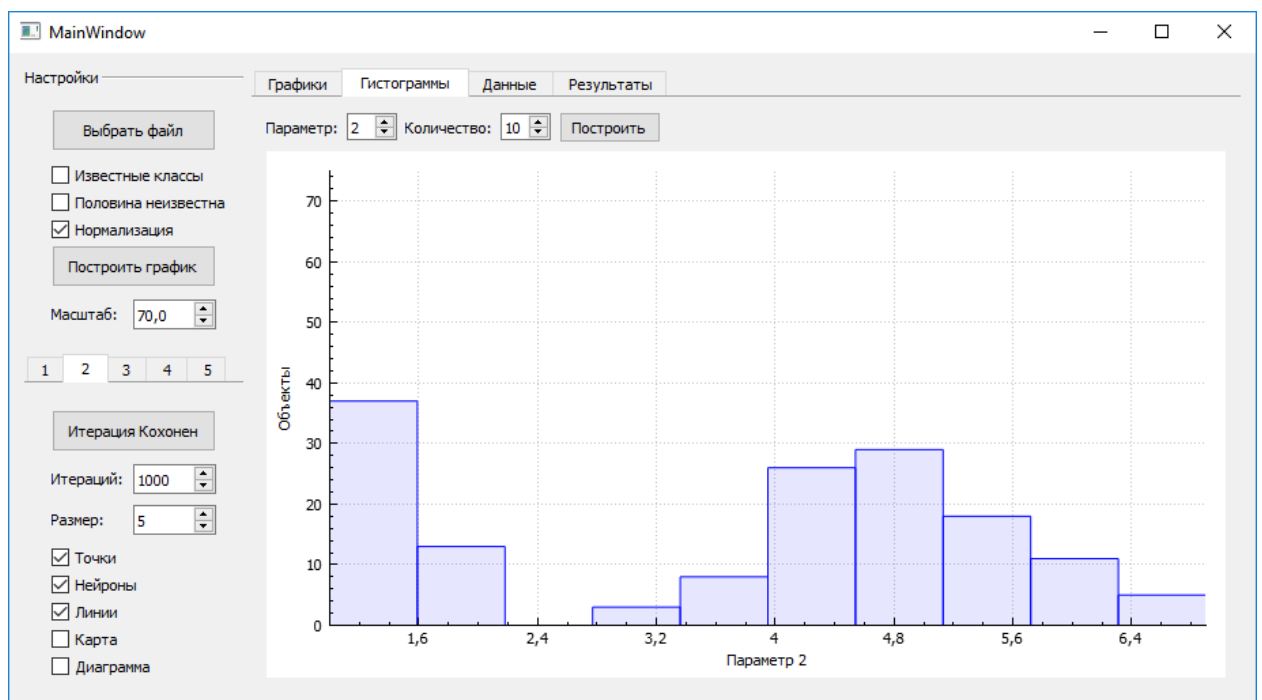


Рисунок 15 – Рабочее окно программы

Перед началом работы с системой необходимо выбрать имеющийся файл из массива данных. При успешной обработке файла и проверки его на наличие ошибок открывается доступ к выбору используемого алгоритма и подстройки его параметров. После выбора оптимальных значений можно запустить алгоритм, сохранить полученные результаты и сравнить работу программы с другими алгоритмами, поменять значения отдельных переменных и запустить повторный анализ данных.

1.6 Выводы

В данном разделе были определены понятие Data Mining, принцип работы, основные аспекты задач классификации, кластеризации и технической диагностики, а также способы их решения. Были рассмотрены задачи, решаемые в ходе выполнения работы: срезы данных, описание атрибутов. Был обоснован выбор программных средств разработки.

2 Изучение методов, выявление недостатков

2.1 Метод k ближайших соседей

Описание алгоритма

Метод k ближайших соседей (англ. k-nearest neighbors algorithm, k-NN) – простейший метрический алгоритм для классификации объектов. Предполагается, что существует определенный набор объектов с уже имеющейся классификацией. Рассмотрим принцип работы алгоритма.

Классифицируемый объект X относится к тому классу Y, к которому принадлежит большинство из k его ближайших соседей.

Входные данные для алгоритма следующие:

- классифицируемая выборка;
- классифицированная выборка;
- параметр k – натуральное число, определяющее максимальное количество соседей, используемых для классификации нового объекта;
- параметр R – радиус поиска соседей.

Приведем пошаговое описание алгоритма:

- а) выбрать случайный неклассифицированный объект выборки X;
- б) найти k ближайших классифицированных объектов для X, находящихся на расстоянии меньше R от него;
- в) найти наиболее распространенный класс Y среди найденных объектов;
- г) объект X добавить в класс Y и исключить его из дальнейшего рассмотрения. Переход к пункту а.

Алгоритм завершает свою работу, когда все объекты выборки являются классифицированными.

Исследование алгоритма на модельных данных

Проверим работу алгоритма на следующем примере (рисунок 16). Объект, обозначенный квадратом, необходимо классифицировать на основе имеющихся данных. При значении k в промежутке [1; 5] объект будет классифицирован как треугольник, поскольку большинство его ближайших соседей это треугольники, а при значении большем шести будет определен как круг.

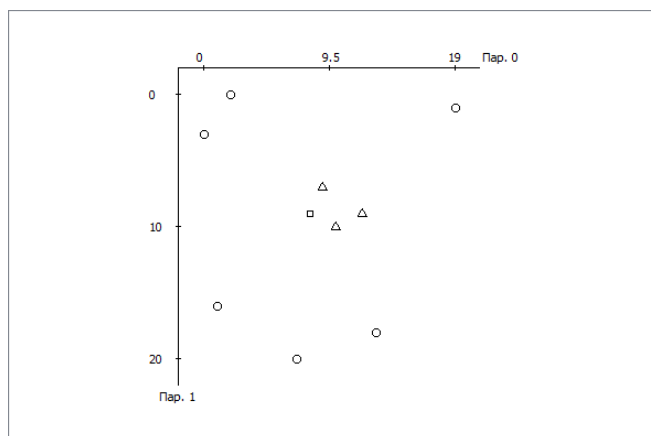


Рисунок 16 – Классифицируемый объект и обучающая выборка

В данном случае имеет смысл ограничить радиус поиска ближайших соседей, если тестовый квадрат должен быть классифицирован как треугольник.

Выводы

Достоинство – простой в реализации алгоритм.

Недостатки:

- высокая вычислительная трудоемкость;
- подходит только для задач небольшой размерности по количеству классов и переменных.

2.2 Алгоритм FRiS-STOLP

Описание алгоритма

Алгоритм FRiS-СТОЛП используется для нахождения эталонных объектов в каждом имеющемся классе объектов на основе FRiS-функции. Данная функция позволяет не только определить как далеко или близко объекты расположены в пространстве или похожи ли они друг на друга, но и дать количественную оценку ответа на вопрос “по сравнению с чем?”. Данный способ определения сходства позволяет учитывать большее количество факторов при классифицировании объектов.

Входные данные для алгоритма следующие:

- классифицируемая выборка;
- классифицированная выборка.

Приведем пошаговое описание алгоритма:

а) для каждого объекта класса А относительно всех остальных объектов этого же класса находим среднее арифметическое значение оценки

обороноспособности относительно одного ближайшего объекта из других классов. Данная функция сходства имеет следующий вид:

$$C'_{ij/b} = (R_2 - R_1)/(R_2 + R_1), \quad (3)$$

где i – объект класса А,

j – другой объект класса А ($i \neq j$),

b – ближайший объект класса В для i ,

R_1 – расстояние между объектами i и j ,

R_2 – расстояние между объектами i и b ;

б) для каждого объекта класса А находим среднее арифметическое значение оценки толерантности относительно одного ближайшего объекта из других классов относительно всех остальных объектов этого же класса В. Данная функция сходства имеет следующий вид:

$$C''_{qs/i} = (R_2 - R_1)/(R_2 + R_1), \quad (4)$$

где i – объект класса А,

s – ближайший объект класса В для i ,

q – другой объект класса В ($s \neq q$),

R_1 – расстояние между объектами q и s ,

R_2 – расстояние между объектами q и i ;

в) для каждого объекта класса А находим общую оценку эффективности по следующей формуле:

$$F_i = (C'_i + C''_i)/2, \quad (5)$$

где i – объект класса А,

C' – среднее значение обороноспособности объекта i ,

C'' – среднее значение толерантности объекта i ;

г) в качестве столпа для класса А выбираем объект, который имеет наибольшую величину F_i .

д) повторяем шаги а-г для всех остальных классов.

Поиск значений эффективности позволяет наиболее точно определить столпы классов и предотвратить возникновения ошибок: чем выше обороноспособность объекта, тем меньше вероятность появления ошибок первого рода, чем выше толерантность объекта, тем меньше вероятность появления ошибок второго рода.

Первые столпы были выбраны в тех условиях, когда им противостояли ближайшие объекты из других классов. Теперь же в роли этих объектов могут выступать уже известные столпы. Пересчет столпов достаточно провести один или два раза в зависимости от количества имеющихся объектов, впоследствии список столпов практически не меняется.

Определение принадлежности объекта к какому-либо классу можно определять двумя способами:

- объект относится к тому классу, к которому принадлежит ближайший для него столп;
- с учетом расстояния до ближайшего из конкурирующих образов, используя FRiS-функцию.

Исследование алгоритма на модельных данных

На рисунке 17 представлены два класса, а также два столпа, обозначенные крестами. Исследуемый объект в форме квадрата расположен несколько ближе к эталону класса формы круга, однако при использовании FRiS-функции он будет определен как треугольник, поскольку, судя по структуре классов, он является типичным представителем треугольных объектов.

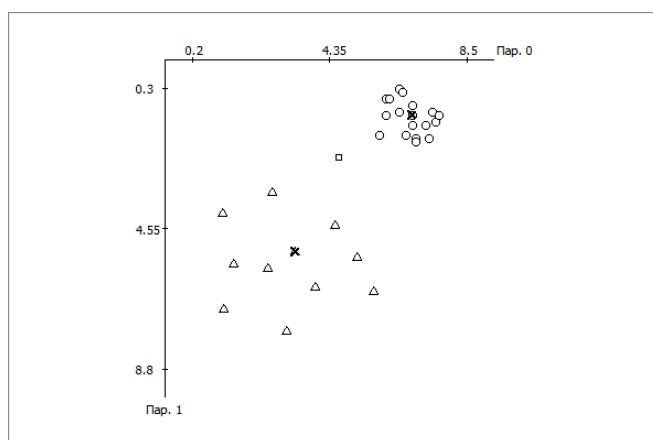


Рисунок 17 – Исходная выборка, столпы и неизвестный объект

Выводы

Достоинства:

- простой процесс распознавания с опорой;
- возможность избавления от ошибок и выбросов.

Недостатки:

- высокая вычислительная трудоемкость;
- подходит только для задач небольшой размерности по количеству классов и переменных.

2.3 Алгоритм FOREL

Описание алгоритма

FOREL (от англ. Formal Element – Формальный Элемент) — алгоритм кластеризации, основанный на идее объединения в один кластер объектов в областях их наибольшего сгущения. Рассмотрим принцип работы алгоритма.

На каждом шаге строится гиперсфера минимального радиуса, которая охватывает все объекты с центром в случайном объекте выборки. Если бы был необходим один кластер, то он был бы представлен именно этой начальной сферой, как отображено на рисунке 18.

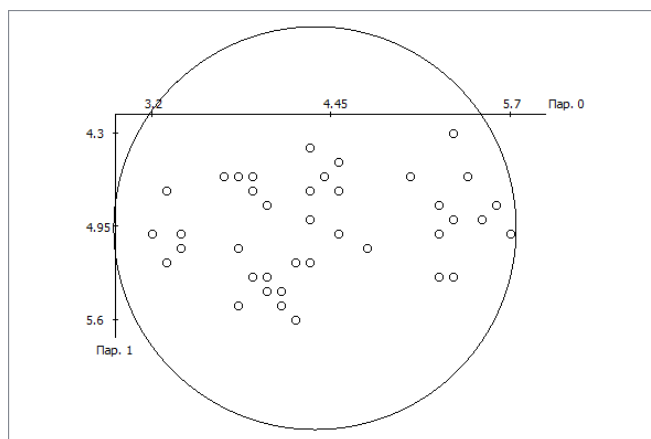


Рисунок 18 – Кластер, содержащий все объекты выборки

Для получения большего числа кластеров необходимо постепенно уменьшать радиус сферы и двигать ее в сторону локального сгущения объектов выборки. Перенос центра сферы с уменьшением радиуса продолжается до тех пор, пока она не остановится, то есть пока на очередном шаге координаты центра не останутся теми же. Этот процесс показан на рисунке 19.

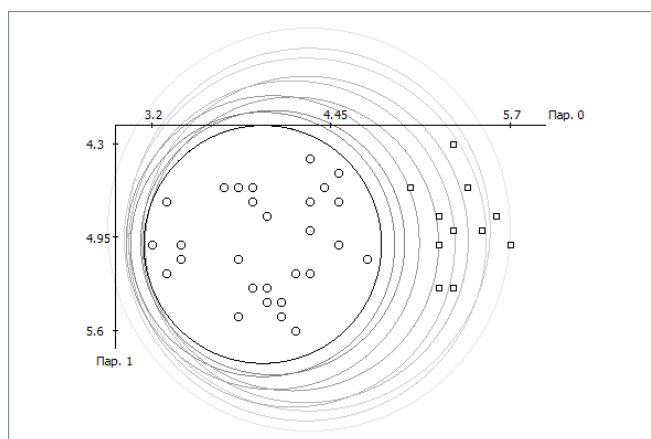


Рисунок 19 – Уменьшение радиуса и перемещение сферы

Полученные объекты внутри сферы помечаются как кластеризованные и убираются из выборки. Этот процесс повторяется до тех пор, пока вся выборка не будет кластеризована.

Входные данные для алгоритма следующие:

- кластеризуемая выборка;
- параметр α – коэффициент в пределах (0; 1) уменьшения радиуса сферы. При меньшем значении получится меньшее количество кластеров.

Приведем пошаговое описание алгоритма:

- а) выбрать случайный объект выборки;
- б) провести сферу минимального радиуса с центром в данном объекте, охватывающую все объекты выборки;
- в) уменьшить радиус сферы по формуле

$$R = R * \alpha, \tag{6}$$

где R – радиус сферы;

α – коэффициент уменьшения радиуса сферы;

г) пометить объекты, находящиеся внутри сферы;

д) передвинуть центр сферы в центр масс помеченных объектов;

е) если центр сферы изменился, то переходим к пункту в, иначе к пункту ж;

ж) помеченные объекты добавить в один кластер и исключить их из дальнейшего рассмотрения. Переход к пункту а.

Алгоритм завершает свою работу, когда все объекты выборки являются кластеризованными.

Исследование алгоритма на модельных данных

На рисунках 20 и 21 можно заметить, что результат кластеризации сильно зависит от выбора объекта, с которого начинается построение начальной сферы.

Во избежание возникновения таких случаев в программной реализации алгоритма в качестве начального объекта выбирался тот, который наиболее близко расположен к центру масс некластеризованных объектов, что позволяет добиться одинаковой кластеризации при повторных запусках алгоритма.

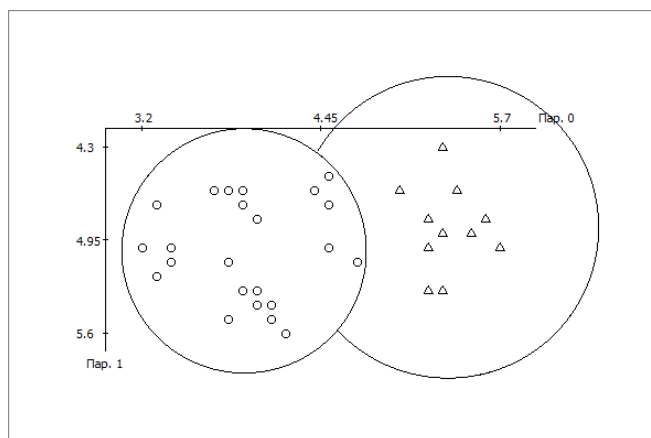


Рисунок 20 – Результат кластеризации

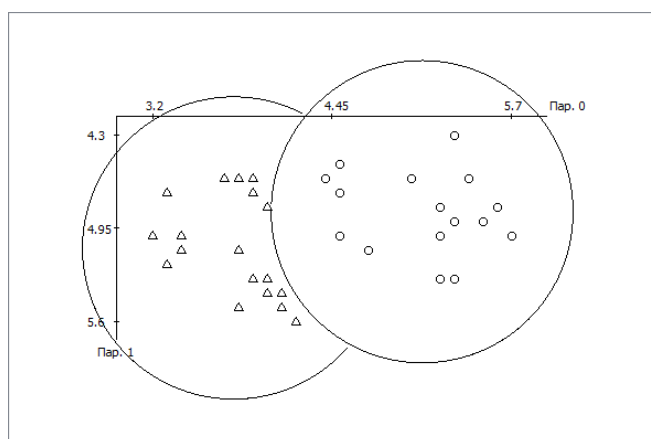


Рисунок 21 – Результат кластеризации

Рассмотрим работу алгоритма на наборе объектов, отображенных на рисунке 22. Визуально оценив расположение объектов, можно распределить их по трем кластерам.

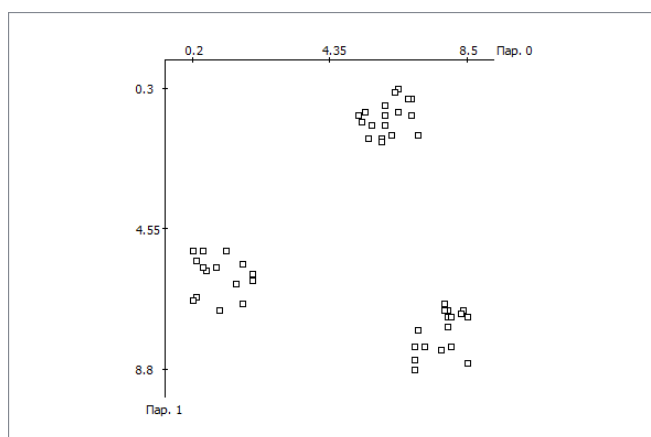


Рисунок 22 – Кластеризуемая выборка

Однако, варьируя коэффициент α , можно получить различное количество кластеров. На рисунке 23 показан график зависимости

коэффициента α и количества кластеров, полученных в результате работы алгоритма.

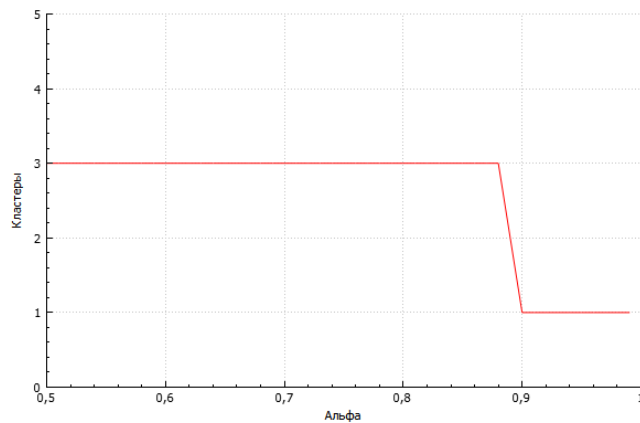


Рисунок 23 – График зависимости α и количества кластеров

Результаты кластеризации, соответствующие данному графику, с использованием разных параметров можно увидеть на рисунках 24-26.

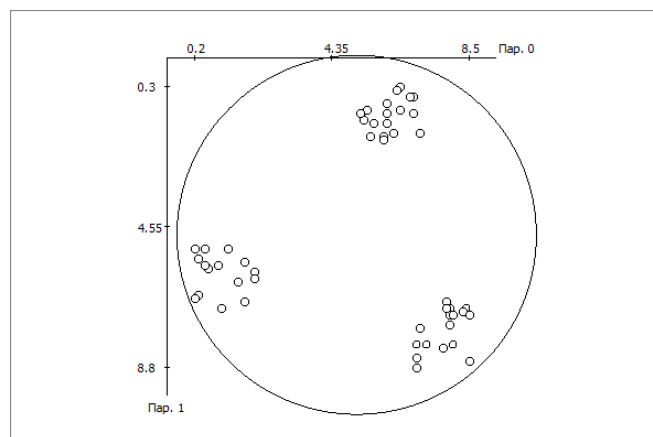


Рисунок 24 – Результат кластеризации при $\alpha = 0.9$

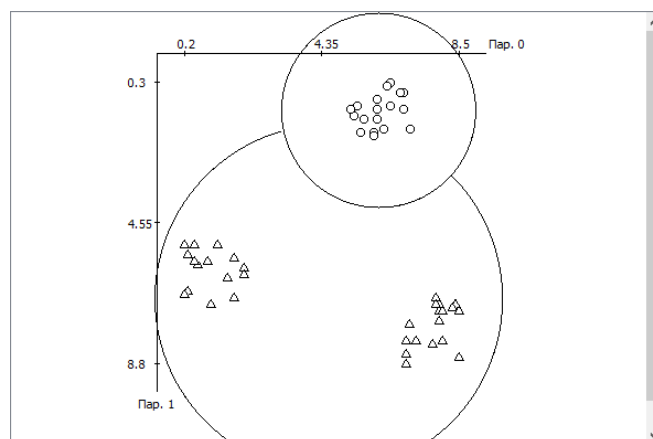


Рисунок 25 – Результат кластеризации при $\alpha = 0.89$

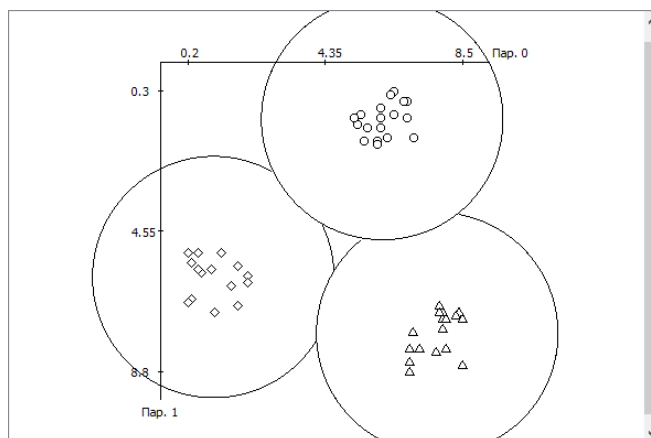


Рисунок 26 – Результат кластеризации при $\alpha = 0.7$

Проверим работу алгоритма, добавив к исходной выборке еще один отдельный объект, расположенный между кластерами. Как видно из графика на рисунке 27, желаемый результат с 4 кластерами получается при использовании параметра α в промежутке от 0.64 до 0.77.

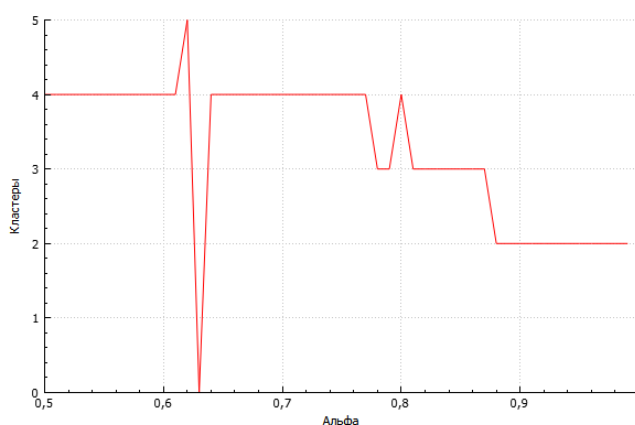


Рисунок 27 – График зависимости α и количества кластеров

Несмотря на то, что в результате работы программы получится 4 кластера, их состав может не соответствовать ожиданиям (рисунок 28).

Такое расположение кластеров вызвано тем, что некоторая часть объектов кластера размещена довольно близко друг к другу, а добавленный новый объект немного притягивает центр сферы к себе. Чтобы предотвратить возникновение таких результатов, необходимо ввести дополнительные модификации алгоритма или использовать другое значение α (рисунок 29).

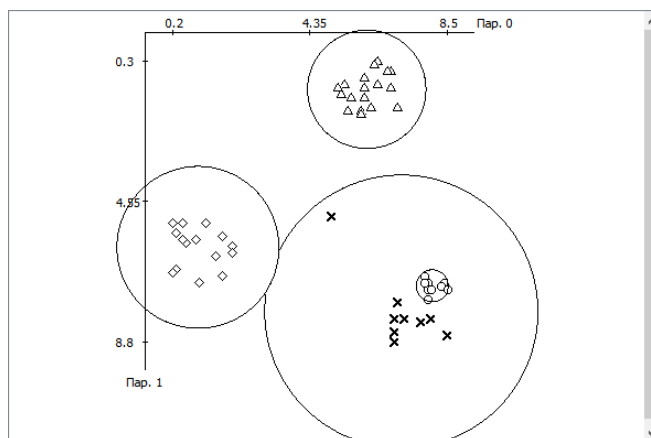


Рисунок 28 – Результат кластеризации при $\alpha = 0.77$

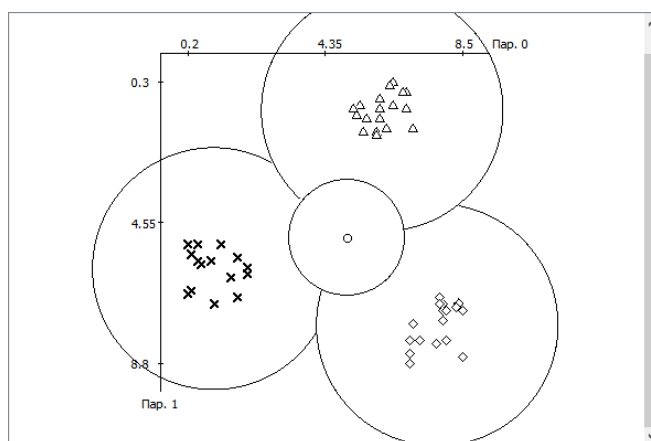


Рисунок 29 – Результат кластеризации при $\alpha = 0.7$

Также на графике имеется такое значение α , при котором количество кластеров равно нулю. Причину возникновения этого случая можно рассмотреть на рисунке 30. При уменьшении радиуса сферы на каждой итерации состав ее внутренних объектов постоянно меняется, вследствие чего сфера уменьшается до такого размера, что перестает содержать в себе какие-либо элементы. Поэтому имеет смысл взять одно из предыдущих состояний сферы либо увеличивать параметр α в процессе работы алгоритма.

Выводы

Достоинства:

- высокое сходство объектов кластеров с центральным элементом;
- сходимость алгоритма за конечное число шагов.

Недостатки:

- произвольное по количеству разбиение на классы;
- плохая применимость алгоритма при близко расположенных кластерах;
- полученные кластеры имеют сферическую форму;

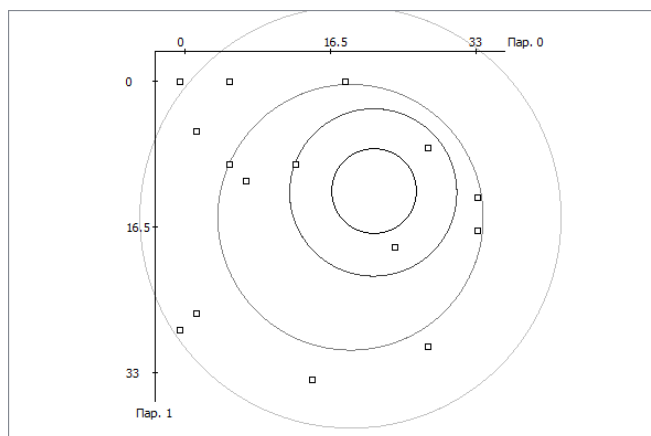


Рисунок 30 – Ошибка при кластеризации

- неустойчивость алгоритма (зависимость от выбора начального объекта);
- необходимость предварительной нормализации значений выборки;
- заикливание алгоритма при малом значении α .

2.4 Метод k-средних

Описание алгоритма

Метод k-средних (англ. – k-means) – один из методов кластеризации, основанный на вычислении центра масс для главных объектов кластеров.

Входные данные для алгоритма следующие:

- кластеризуемая выборка;
- параметр α – количество результирующих кластеров.

Приведем пошаговое описание алгоритма:

- создать α главных объектов выборки, присвоить им случайные значения;
- для каждого главного объекта вычислить центр масс;
- перенести все главные объекты в найденные центры масс;
- если хоть один из главных объектов сдвинулся, то переход к пункту б, иначе алгоритм завершает свою работу.

Исследование алгоритма на модельных данных

Рассмотрим работу алгоритма на наборе объектов, отображенных на рисунке 31. Добавим 3 главных объекта, которые будут характеризовать центры будущих кластеров и поместим их в случайные объекты выборки.

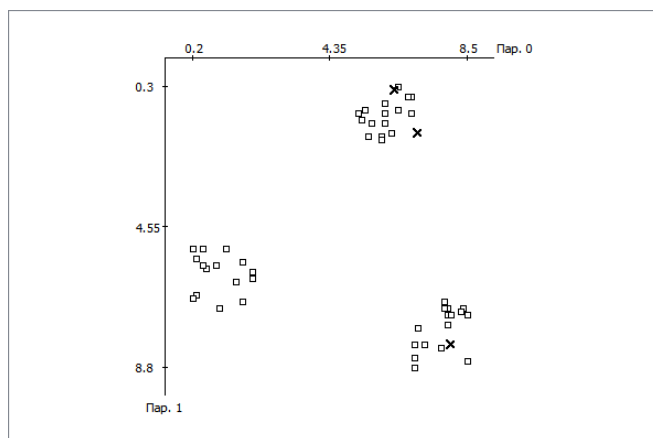


Рисунок 31 – Исходная выборка и три главных объекта

После проведения нескольких итераций алгоритма главные объекты сдвигаются к локальным сгущениям элементов выборки (рисунок 32). Таким образом, каждый из объектов выборки будет принадлежать к ближайшему для него главному объекту, который и образует отдельный кластер.

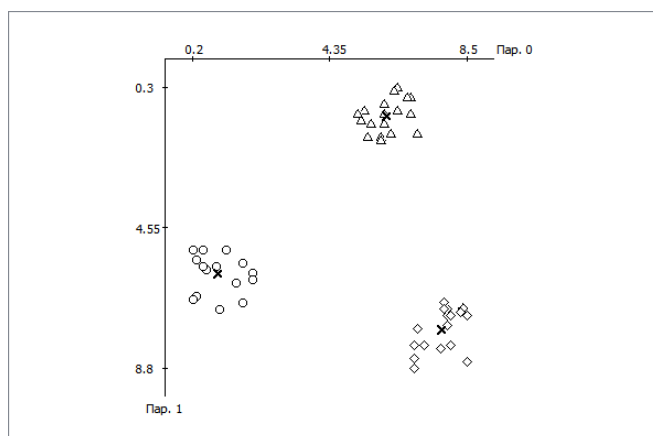


Рисунок 32 – Результат кластеризации

Выбор начальных объектов случайным образом позволяет получить различные варианты кластеризации, пример одной из вариаций можно увидеть на рисунке 33.

Выводы

Достоинства:

- разбиение на заранее известное количество кластеров;
- возможность избавления от ошибок и выбросов.

Недостатки:

- результат кластеризации зависит от выбора исходных центров кластеров;
- подходит только для задач небольшой размерности по количеству классов и переменных.

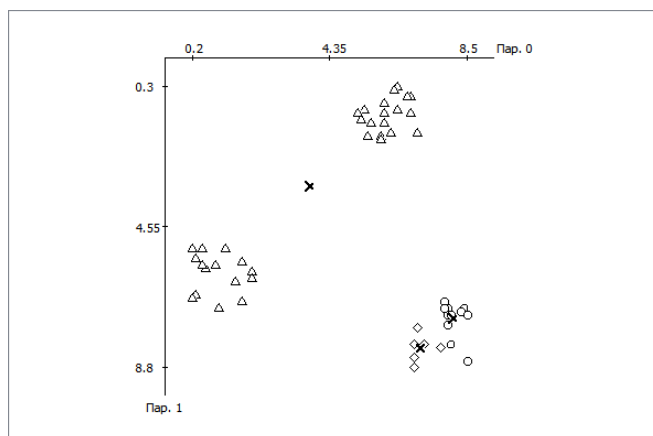


Рисунок 33 – Результат кластеризации

2.5 Кластеризатор на основе нейронной сети Кохонена

Описание алгоритма

Самоорганизующаяся нейронная сеть (карта) Кохонена выполняет проецирование многомерных данных в пространство меньшей размерности. Она представляет собой двухслойную сеть (рисунок 34). Каждый нейрон первого (входного) слоя соединен со всеми нейронами второго (выходного) слоя, которые расположены в виде двумерной решетки. Количество входных нейронов совпадает с размерностью кластеризуемых данных.

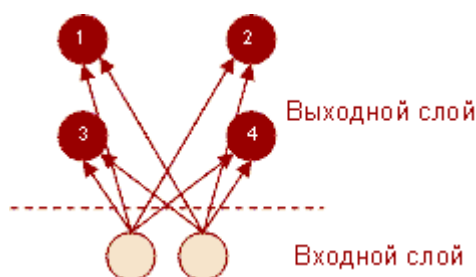


Рисунок 34 – Два слоя сети Кохонена

Нейроны выходного слоя называются кластерными элементами, их количество определяют максимальное количество групп, на которые система может разделить входные данные. Увеличивая количество нейронов второго слоя можно увеличивать детализацию результатов процесса кластеризации.

При визуализации данных таким способом можно увидеть наличие или отсутствие кластерной структуры данных, число кластеров, законы совместного распределения признаков, зависимости между переменными. Рассмотрим принцип работы алгоритма.

Система работает по принципу соревнования – нейроны второго слоя соревнуются друг с другом за право наилучшим образом сочетаться с входным вектором сигналов. На каждом шаге обучения из исходного набора данных случайно выбирается один вектор (объект). Затем производится поиск нейрона выходного слоя, для которого расстояние между его вектором весов и входным вектором минимально. По определённому правилу производится корректировка весов для нейрона-победителя и нейронов из его окрестности (рисунок 35), которая задаётся соответствующей функцией окрестности.

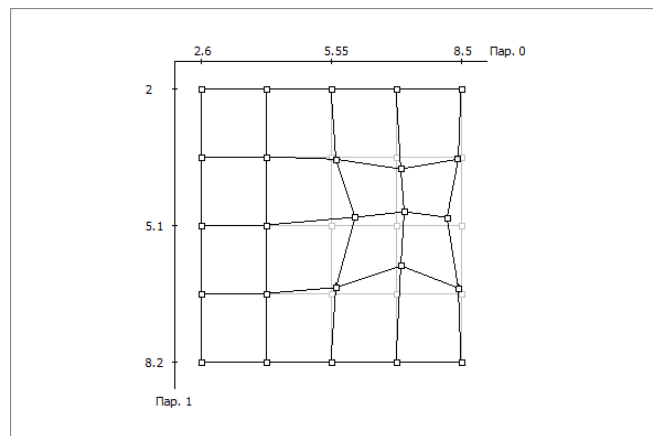


Рисунок 35 – Корректировка весов для нейрона-победителя и его соседей

Радиус действия функции окрестности будет уменьшаться с увеличением количества итераций. В качестве такой функции была использована функция Гаусса, плотность распределения которой показана на рисунке 36. Функция Гаусса имеет вид:

$$h(u, c, i) = \exp\left(-\frac{\rho(c, u)}{\sigma(i)}\right), \quad (7)$$

где u – номер нейрона в двумерной решетке второго слоя сети, для которого вычисляем значение h ;

c – номер нейрона-победителя в двумерной решетке второго слоя сети;

i – номер текущей итерации,

$$\sigma(i) = \frac{1}{\exp(i^{-2})}, \quad (8)$$

где i – номер текущей итерации.

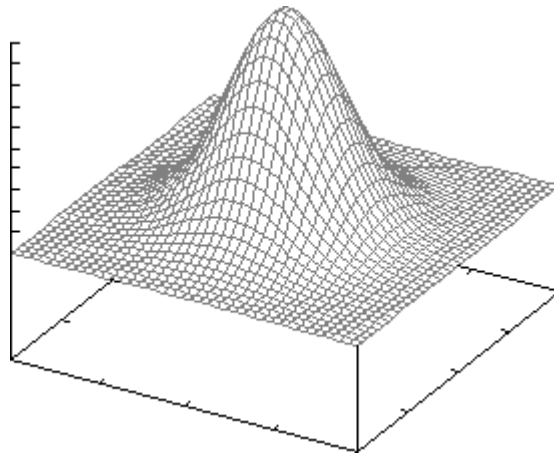


Рисунок 36 – Функция Гаусса

Входные данные для алгоритма следующие:

- кластеризуемая выборка;
- параметр N – размерность сети $N \times N$ в пределах $[1; 10]$, при меньшем значении получится меньшее количество кластеров;
- параметр I – количество выполняемых операций;
- параметр η – коэффициент скорости обучения в пределах $(0; 1)$.

Приведем пошаговое описание алгоритма:

- а) инициализировать матрицу весов случайными значениями в пределах $[-1; 1]$;
- б) установить счетчик итераций i равным 0;
- в) выбрать случайный объект выборки x ;
- г) найти нейрона-победителя c для данного объекта;
- д) изменить вес каждого нейрона w_u по формуле:

$$\overline{w}_u = \overline{w}_u + (\overline{w}_u - \bar{x}) * h(u, c, i) * \eta, \quad (9)$$

где \overline{w}_u – вектор весовых коэффициентов нейрона с индексом u ;

\bar{x} – вектор весовых коэффициентов объекта выборки;

$h(u, c, i)$ – значение функции окрестности (7) для нейрона с номером u и нейрона-победителя с номером c на i -ой итерации;

η – коэффициент скорости обучения;

е) увеличить счетчик итераций i на единицу, переход к пункту в.

Алгоритм завершает свою работу после выполнения I итераций.

Для остановки процесса обучения также можно использовать следующие критерии:

- выход сети стабилизируется, т.е. входные вектора не переходят между кластерными элементами;
- изменения весов становятся незначительными.

Исследование алгоритма на модельных данных

В программной реализации алгоритма для оптимизации его работы в качестве начальных значений матрицы весов выходного слоя использовались промежуточные значения между минимальным и максимальным значением входных векторов. Таким образом, карта получается натянутой на крайние элементы кластеризуемой выборки в соответствии с рисунками 37 и 38.

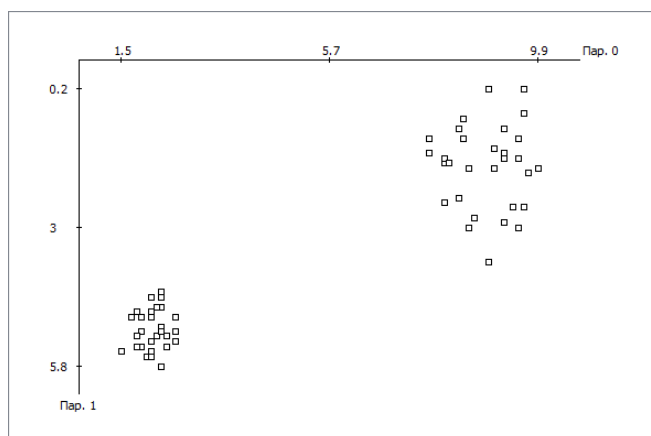


Рисунок 37 – Кластеризуемая выборка

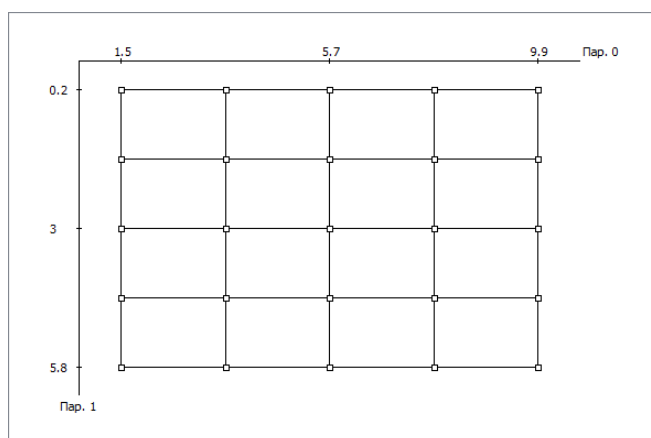


Рисунок 38 – Сеть Кохонена

Для того чтобы можно было провести кластерный анализ, проведем 1000 итераций обучения сети. На рисунке 39 видно, что некоторые нейроны заметно сдвинулись со своего изначального положения в сторону локального сгущения множества объектов, поскольку во время работы алгоритма они неоднократно выступали в качестве нейронов-победителей.

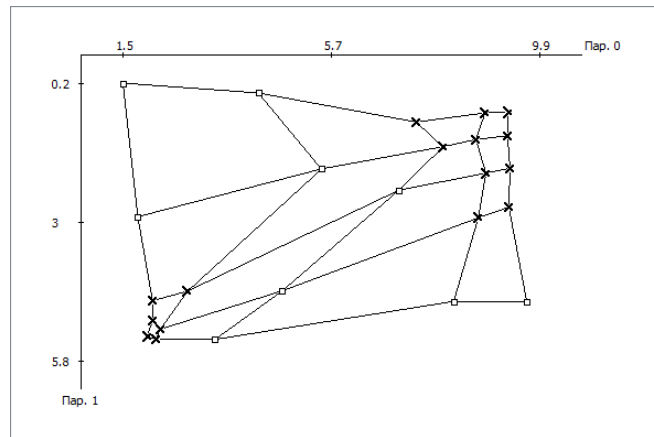


Рисунок 39 – Сеть Кохонена, 1000 итераций обучения

Однако, несмотря на проделанную работу, часть нейронов осталась практически на том же месте. Проведя еще 9000 итераций, можно заметить, что сеть уже является сформированной (рисунок 40): при проведении дополнительных итераций изменения весов нейронов становятся незначительными.

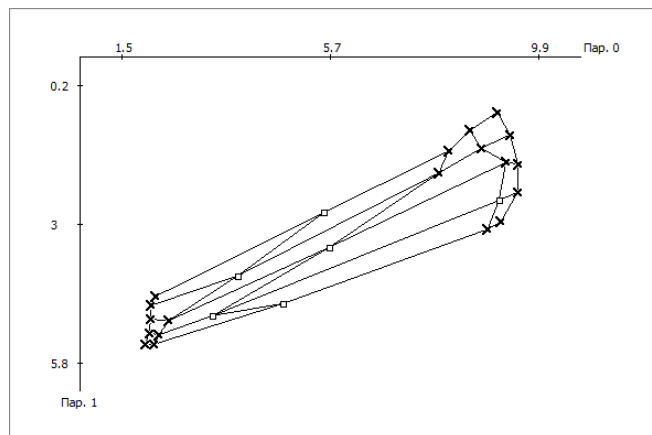


Рисунок 40 – Сеть Кохонена, 10000 итераций обучения

В данный момент нейроны, обозначенные крестом, являются нейронами-победителями для определенного набора объектов и образуют вместе с ними отдельный кластер. Построим на основе выходного слоя нейронов диаграмму, отображающую распределение объектов по кластерам (рисунок 41).

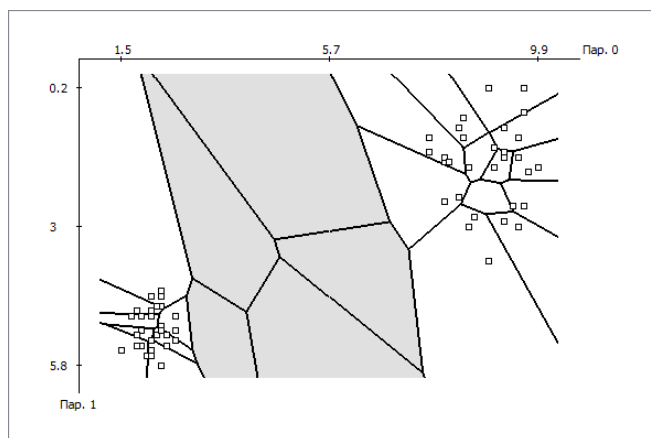


Рисунок 41 – Распределение исходной выборки

Из данного рисунка видно, что области, закрашенные серым цветом, не содержат в себе ни единого объекта, а, следовательно, не образуют никаких кластеров. Кластеры, образованные остальными нейронами, при необходимости можно объединить в один поиском в ширину или другими способами.

Выводы

Достоинства:

- высокая точность работы алгоритма;
- результирующие кластеры не пересекаются и имеют произвольную форму;
- наглядное представление многомерных данных на плоскости;
- регулируемое по количеству разбиение на классы;
- наличие эталонного объекта кластеров.

Недостатки:

- сложность алгоритма и медленная скорость работы;
- необходимость введения алгоритма объединения отдельных нейронов;
- наличие нейронов-победителей, расположенных между отдельными предполагаемыми кластерами.

Сами по себе нейронные сети задачу кластерного анализа и идентификации зависимостей не решают. Они только позволяют по внешнему виду карты выдвинуть гипотезы о наличии кластерной структуры и числе кластеров, зависимостях между значениями отдельных переменных. Выдвинутые гипотезы должны проверяться и подтверждаться иными способами. Более того, карты Кохонена могут приводить как к формированию ложных гипотез, так и к невозможности увидеть отдельные реально имеющиеся и статистически достоверные зависимости в данных.

2.6 Выводы

В данном разделе были исследованы особенности разработанных алгоритмов. Даны краткие описания алгоритмов, используемые параметры, коэффициенты и входные данные. Описан порядок выполнения алгоритмов, приведены примеры работы алгоритмов на модельных данных с иллюстрациями, определены основные достоинства и недостатки рассматриваемых методов.

3 Численные исследования

3.1 Задача 1 – «Ирисы Фишера»

В рамках выполнения выпускной квалификационной работы было предложено рассмотреть работу нескольких алгоритмов классификации на наборе данных с ирисами.

На рисунках 42 и 43 представлены исходные данные для алгоритмов. На рисунках 44–48 представлены результаты работы алгоритмов. Результаты исследований представлены в таблице 2.

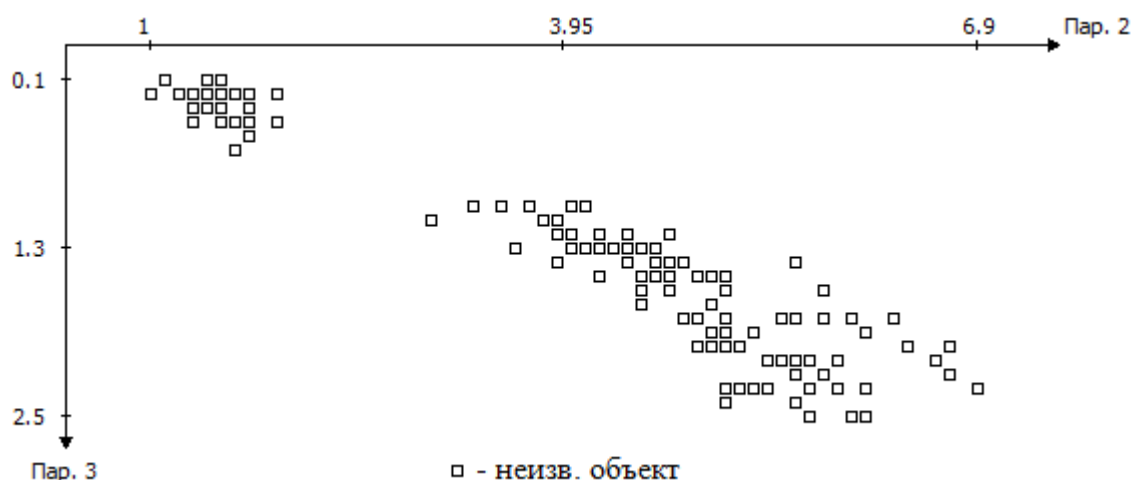


Рисунок 42 – Исходные данные для задачи кластеризации

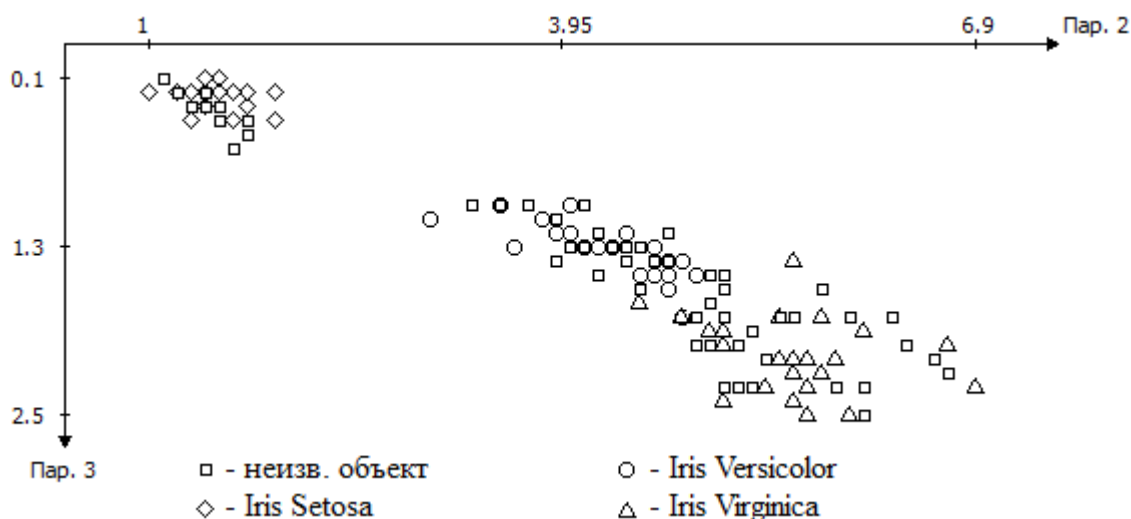


Рисунок 43 – Исходные данные для задачи классификации

Метод k ближайших соседей

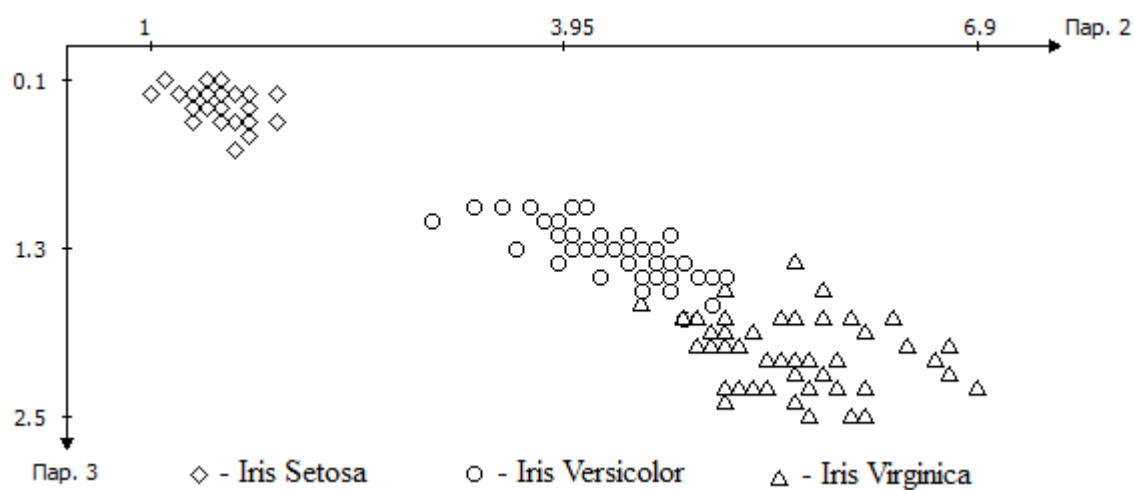


Рисунок 44 – Результат классификации методом k ближайших соседей

Алгоритм FRiS-STOLP

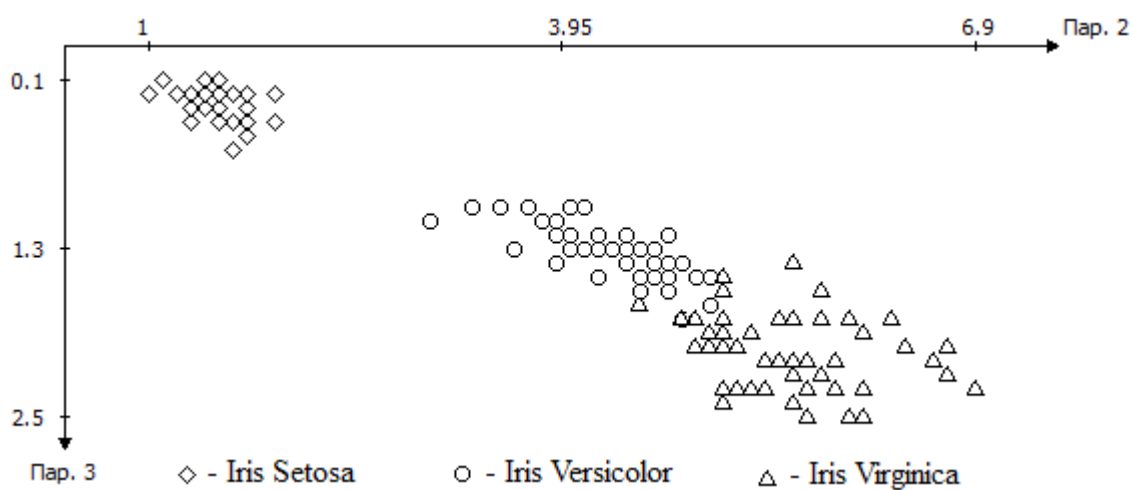


Рисунок 45 – Результат классификации алгоритмом FRiS-STOLP

Алгоритм FOREL

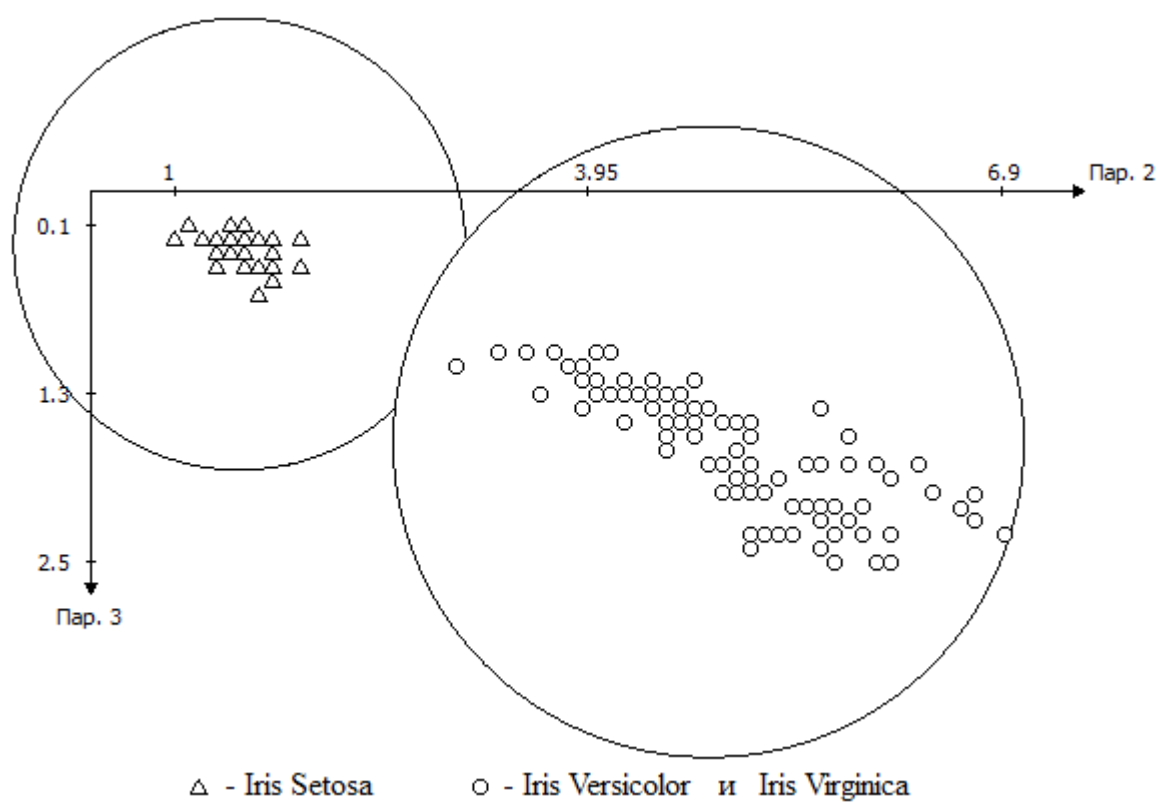


Рисунок 46 – Результат кластеризации алгоритмом FOREL

Метод k-средних

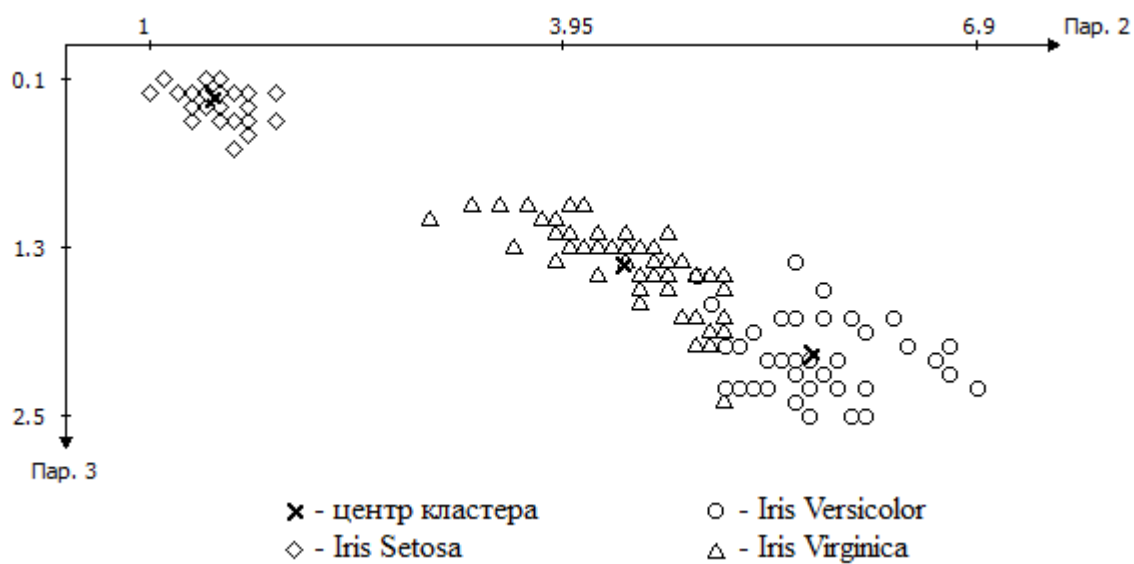


Рисунок 47 – Результат кластеризации методом k-средних

Сеть Кохонена

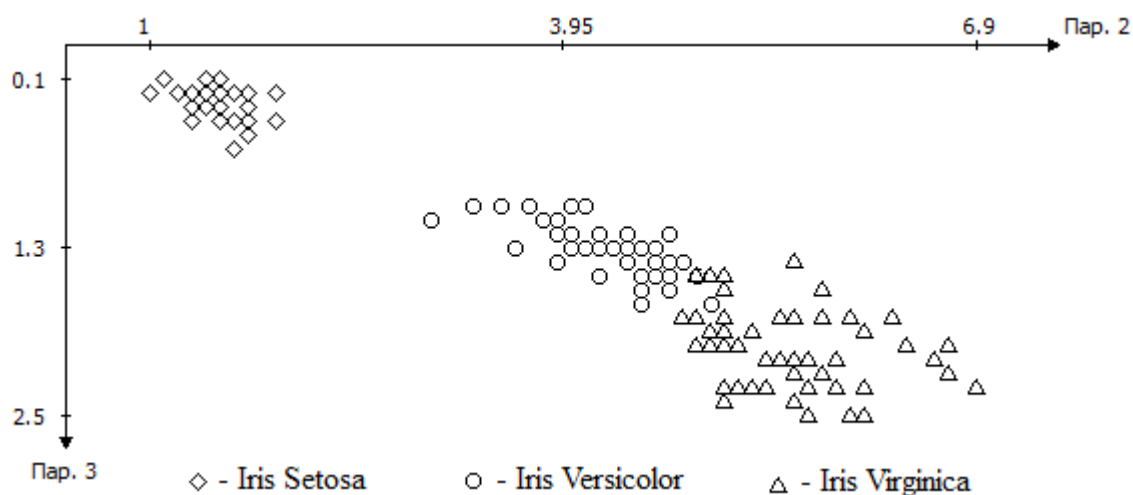


Рисунок 48 – Результат кластеризации при использовании нейронной сети

Таблица 2 – Результаты исследований

Алгоритм	Исх. данные	Результат	
		объекты	верно определено
к ближ. соседей	Неизвестно – 75 объектов 1 класс – 25 объектов 2 класс – 25 объектов 3 класс – 25 объектов	1 класс – 26 2 класс – 24 3 класс – 25	98.6%
FRiS-STOLP	3 класс – 25 объектов	1 класс – 25 2 класс – 25 3 класс – 25	100%
FOREL	Неизвестно – 150 объектов	1 кластер – 50 2 кластер – 100	66.6%
к-средних		1 кластер – 50 2 кластер – 69 3 кластер – 31	86.6%
Сеть Кохонена		1 кластер – 50 2 кластер – 49 3 кластер – 51	99.3%

3.2 Задача 2 – одна партия ЭРИ

Алгоритм FOREL был исключен из дальнейшего рассмотрения, поскольку он приспособлен к работе с кластерами, имеющие небольшое внутрикластерное расстояние и большое межкластерное. На рисунке 49 представлены исходные данные для алгоритмов. На рисунках 50 и 51 представлены результаты работы алгоритмов. Разработанные алгоритмы

ЗАКЛЮЧЕНИЕ

В работе были исследованы следующие алгоритмы: k -ближайших соседей, FRiS-STOLP, FOREL, k -средних, кластеризатор на основе самоорганизующейся сети Кохонена, а также разработана автоматизированная система для решения задачи распознавания образов.

Алгоритм FOREL плохо работает на реальных данных, поскольку он позволяет выделять только те кластеры, которые расположены на большом расстоянии друг от друга. FRiS-STOLP на основе результатов работы по выборке «Ирисы Фишера» дает более высокую точность классификации объектов в сравнении с методом k -ближайших соседей. На предоставленных данных по ЭРИ было выделено 3 кластера алгоритмом k -средних и сетью Кохонена, что соответствует действительности. Состав кластеров не может быть сравнен с эталонным, поскольку известно лишь наличие кластерной структуры.

Оценить качество работы на многомерных данных сложно, так как размерность исследуемой выборки высока, а просмотреть срезы данных по всем наборам признаков является физически невозможным. Однако полученные результаты позволяют обозначить элементы определенных кластеров ЭРИ как надежные изделия и потенциально ненадежные.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 09.01.2014. – Красноярск: СФУ, 2014. – 60 с.
2. Лафоре, Р. Объектно-ориентированное программирование в С++. Классика Computer Science. 4-е изд. / Р. Лафоре – СПб.: Питер, 2014. – 928 с.
3. Муравьев, А. С. Модифицированный алгоритм растущего нейронного газа применительно к задаче классификации / А. С. Муравьев, А. А. Белоусов // Вестник науки Сибири. – 2014. – №4(14). – С. 105-111.
4. Муртазаев, А. К. Фазовые переходы в антиферромагнитной модели Изинга на квадратной решетке с взаимодействиями вторых ближайших соседей / А. К. Муртазаев // Журнал экспериментальной и теоретической физики. – 2013. – Т. 144, вып. 6. – С. 1236-1245.
5. Павловская, Т. А. С/С++. Программирование на языке высокого уровня / Т. А. Павловская. – СПб.: Питер, 2012. – 461 с.
6. Попова, О. А. Модели и методы интеллектуального анализа данных: учебно-методическое пособие [Электронный курс] / О. А. Попова. – Красноярск: Сиб. федер. ун-т, 2012.
7. Сараев, В. Отдайтесь большой цифре / В. Сараев // Эксперт. – 2015. – №9. – С. 51-55.
8. Удалова, Ю. В. Математические и алгоритмические основы объектно-ориентированного программирования: [Электронный курс]: учеб.-метод. пособие / Ю. В. Удалова. – Красноярск: Сиб. федер. ун-т, 2013.
9. Фазылов, Ш. Х. Модель распознающих операторов, основанных на принципе ближайшего соседа, в условиях взаимосвязанности признаков / Ш. Х. Фазылов, Чье Ен Ун // Информатика и системы управления. – 2012. – №4(34). – С. 34-42.
10. Царев, Р. Ю. Алгоритмы и структуры данных: учеб. пособие / Р. Ю. Царев. – Красноярск: Сиб. федер. ун-т, 2013. – 160 с.
11. Шаграев, А. Г. Трансдуктивное обучение логистической регрессии в задаче классификации текстов / А. Г. Шаграев, И. А. Бочаров, В. Н. Фальк // Программные продукты и системы. – 2014. – №2. – С. 114-118.

ПРИЛОЖЕНИЕ А

Техническое задание

Предметная область: Система технической диагностики электрорадиодеталей.

Введение

Настоящее техническое задание, оформленное в соответствии с ГОСТ 19.201-78, содержит требования к системе технической диагностики, предназначенной для обработки информации об электрорадиодеталях (далее ЭРИ).

Основания для разработки

Основания для разработки

Основанием для разработки системы диагностики является задание на выпускную квалификационную работу в форме дипломного проекта.

Исполнитель и заказчик

Система диагностики написана в рамках выполнения выпускной квалификационной работы. Кафедра ИС ИКИТ СФУ.

Исполнителем разработки, выполняемой по-настоящему ТЗ, является студент группы КИ12-11Б Хохлов Александр Николаевич.

Наименование

Программе, разрабатываемой по-настоящему ТЗ, присваивается наименование: «Система технической диагностики ЭРИ», в дальнейшем по тексту именуемая СТД.

Назначение разработки

СТД предназначена для следующие действий:

- обработки информации о пациенте;
- обработка данных большой размерности;
- возможность прогнозирования изменений медицинских показателей пациента.

Требования к программе и программному изделию

Требования к составу

СТД должна состоять из одного модуля, выполняющего все требуемые функции.

Требования к составу выполняемых функций

СТД должна выполнять следующие функции

- обрабатывать информацию о пациенте;
- обработка данных большой размерности;
- выводить полученные результаты в текстовом и графическом видах;

- выводить полученные результаты в текстовый файл.

Требования к надежности

Надежность СТД должна быть обеспечена правильностью алгоритмических решений и программирования.

Условия эксплуатации

СТД должна функционировать в соответствии с заданными в настоящем ТЗ требованиями, в составе ПО ПЭВМ, при эксплуатации ПЭВМ. Условия эксплуатации должны соответствовать условиям эксплуатации ПЭВМ, требования к которым предъявляются в эксплуатационной документации ПЭВМ.

Требования к составу и параметрам технических средств

ИС должна функционировать на ПЭВМ со следующими характеристиками:

- процессор не ниже Pentium II 500МГц;
- объем ОЗУ не менее 256 Мб;
- НЖМД не менее 4 Гб;
- графический адаптер не хуже SVGA 8 Мб;
- монитор не хуже SVGA 0.26, 15 дюймов;
- манипулятор типа «мышь»;
- манипулятор типа «клавиатура».

Требования к информационной и программной совместимости

В качестве языков программирования СТД должен быть использован язык программирования С++.

АСУ должна функционировать на ПЭВМ с операционной системой Windows Vista, 7, 8, 10.

Для реализации интерфейса между СТД и пользователем должны использоваться средства графического интерфейса операционной системы.

Для реализации интерфейса между СТД и другими программами из состава ПО ПЭВМ должны использоваться средства буфера обмена операционной системы.

Требования к маркировке и упаковке

Маркировка НГМД с ИС должна проводиться в соответствии с требованиями ГОСТ 19.102-77 («ЕСПД. Стадии разработки программ и программной документации») ЕСПД.

Требования к транспортировке и хранению

Требования к транспортировке

ИС должна транспортироваться:

- в составе ПЭВМ, записанный на НЖМД ПЭВМ;

- на НГМД.

Условия транспортировки СТД в составе ПЭВМ должны соответствовать условиям транспортировки ПЭВМ, требования к которым предъявляются в эксплуатационной документации ПЭВМ или ее составных частей.

Условия транспортировки СТД на НГМД должны соответствовать условиям транспортировки НГМД, требования к которым предъявляются в эксплуатационной документации НГМД.

Требования по хранению

СТД должна храниться:

- в составе ПЭВМ, записанный на НЖМД ПЭВМ;
- на НГМД.

Условия хранения СТД в составе ПЭВМ должны соответствовать условиям хранения ПЭВМ, требования к которым предъявляются в эксплуатационной документации ПЭВМ или ее составных частей.

Условия хранения СТД на НГМД должны соответствовать условиям хранения НГМД, требования к которым предъявляются в эксплуатационной документации НГМД.

Специальные требования

Требования не предъявляются.

Требования к программной документации

Требования к составу документации

Состав документации определяется Исполнителем на этапе разработки перечнем разрабатываемых документов и согласовывается с Заказчиком.

В комплект документации в обязательном порядке должны входить:

- спецификация;
- текст программы;
- руководство оператора;
- загрузочные модули;
- программа и методики испытаний.

Требования к оформлению документации

Программная документация должна быть разработана и оформлена в соответствии с ЕСПД.

Технико-экономические требования

Трудоемкость разработки, отладки и испытаний СТД должна быть согласована Исполнителем и Заказчиком на этапе формирования задания на выполнение работ.

Стадии и этапы разработки

Создание и отработка СТД должно производиться по следующим основным этапам:

- а) разработка программы и программной документации;
- б) испытание программы на модельных данных;
- в) испытание программы на реальных данных;
- г) работа с программой.

Порядок внесения изменений в техническое задание

Настоящее техническое задание может уточняться и дополняться в процессе создания системы совместными решениями сторон, подписавших техническое задание, оформленными в виде дополнения к ТЗ в соответствии с требованиями ГОСТ 19.201-78.